



Education

Object-Based File Systems: An Overview

Craig Harmer
Symantec

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
 - ◆ Any slide or slides used must be reproduced in their entirety without modification
 - ◆ The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

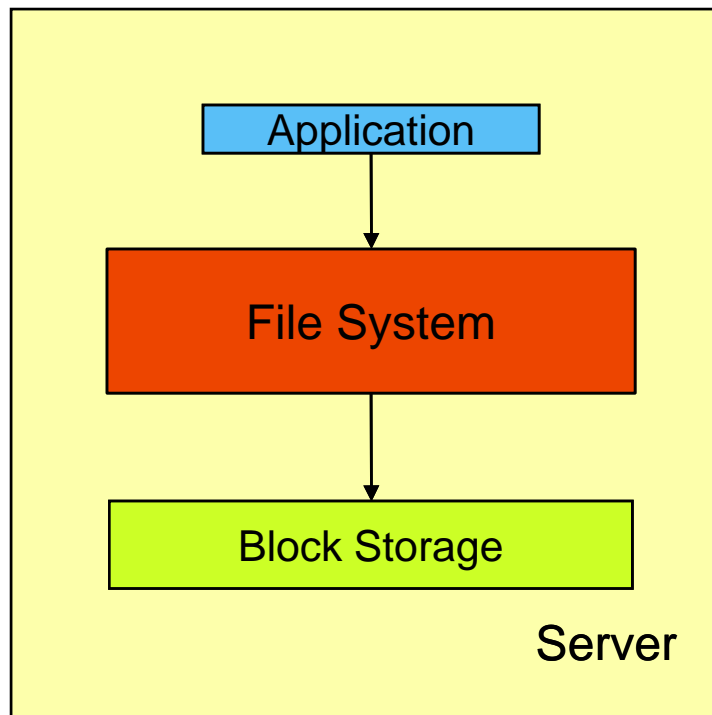
➤ Object-Based File Systems: An Overview

- ◆ Object Storage Devices (OSDs) have been well-publicized, and storage systems that incorporate them are available in the market. OSDs enable more scalable storage systems, massive throughput, and at the same time, enhance data security. To fully deliver on their promise, however, they must be integrated into a file system that provides a usable namespace.
- ◆ This tutorial will briefly review the properties of OSDs, and describe how these new properties are used to rearchitect traditional file system designs to provide scaling, resiliency, and cost benefits compared to traditional file systems.

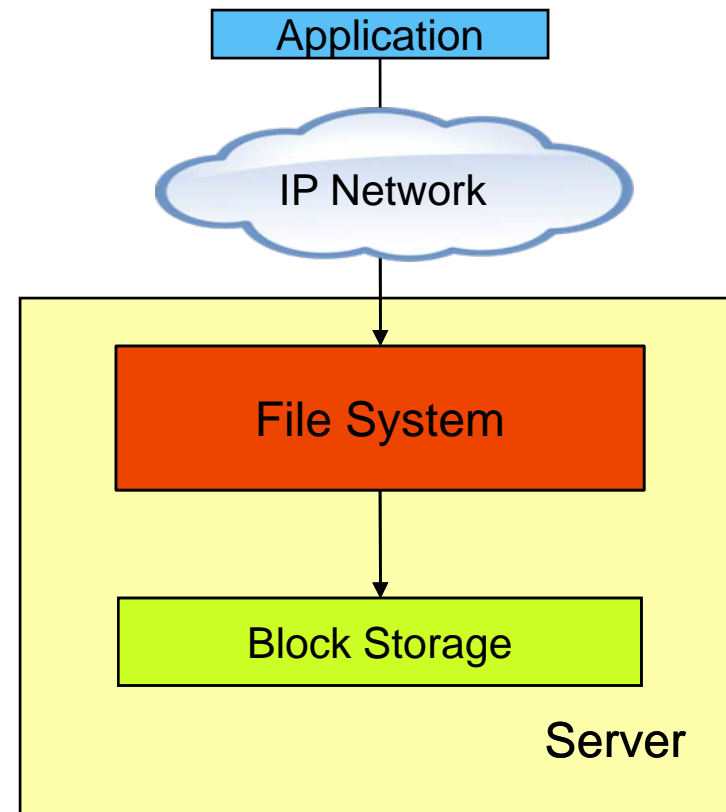
- Motivation
 - ◆ The evolution of storage
- What is an OSD (Object Storage Device)?
 - ◆ Intro
 - ◆ Details
- Building File Systems from OSDs
 - ◆ Simple Implementation
 - ◆ Adding Striping for Performance
 - ◆ Adding Redundancy for Reliability
- Production Quality Object-Based File Systems

Motivation

Direct Attached Storage

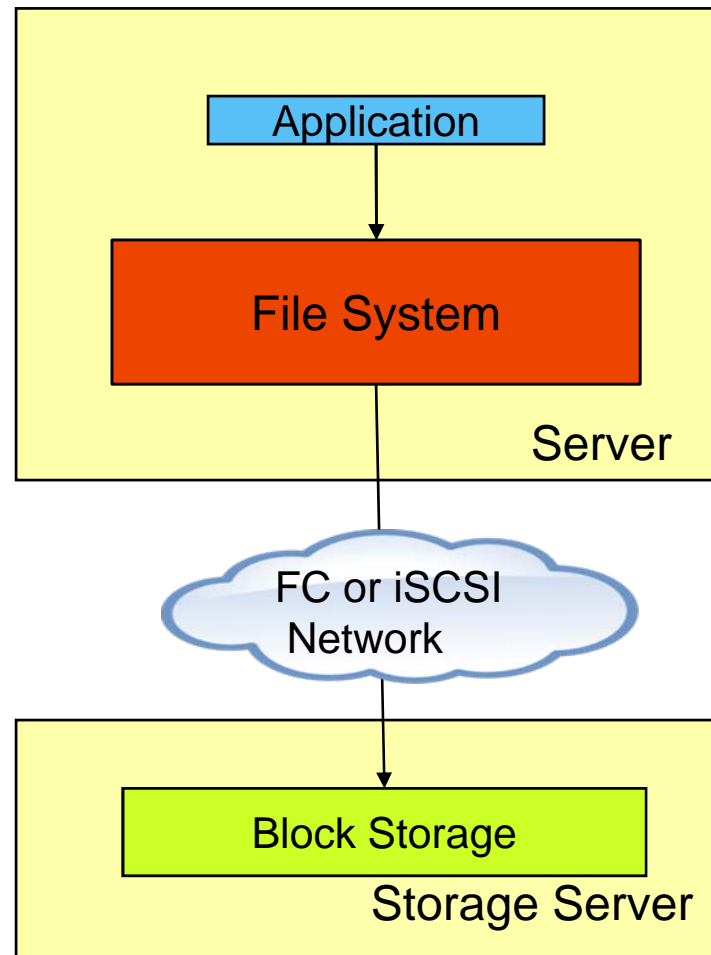


Network Attached Storage

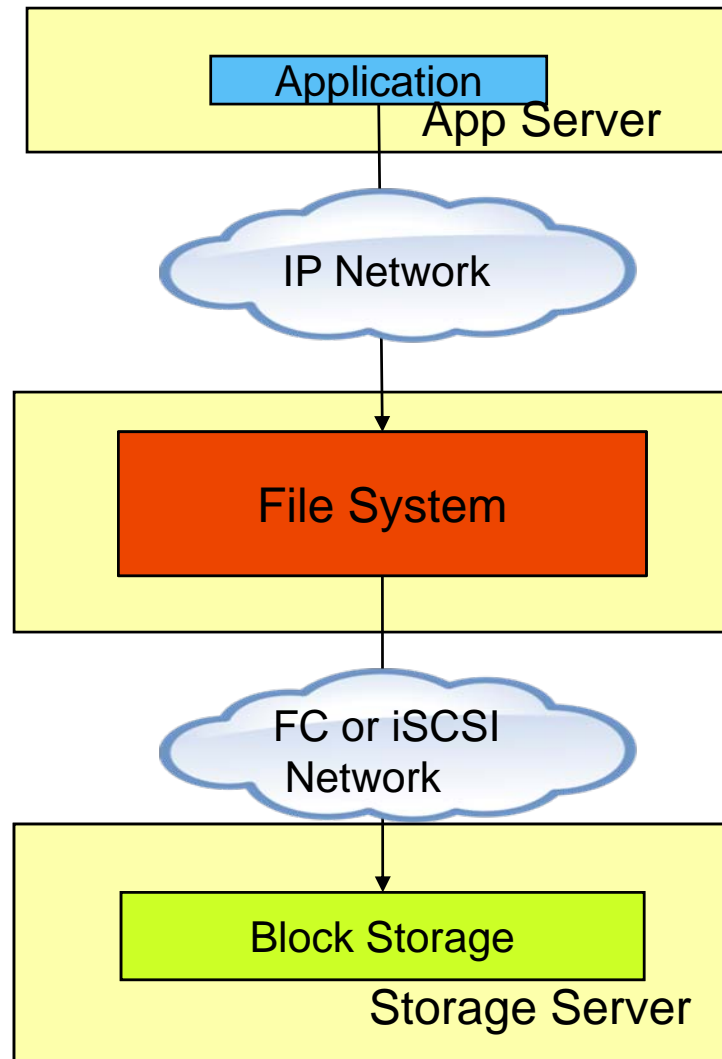


... to Storage Area Networks

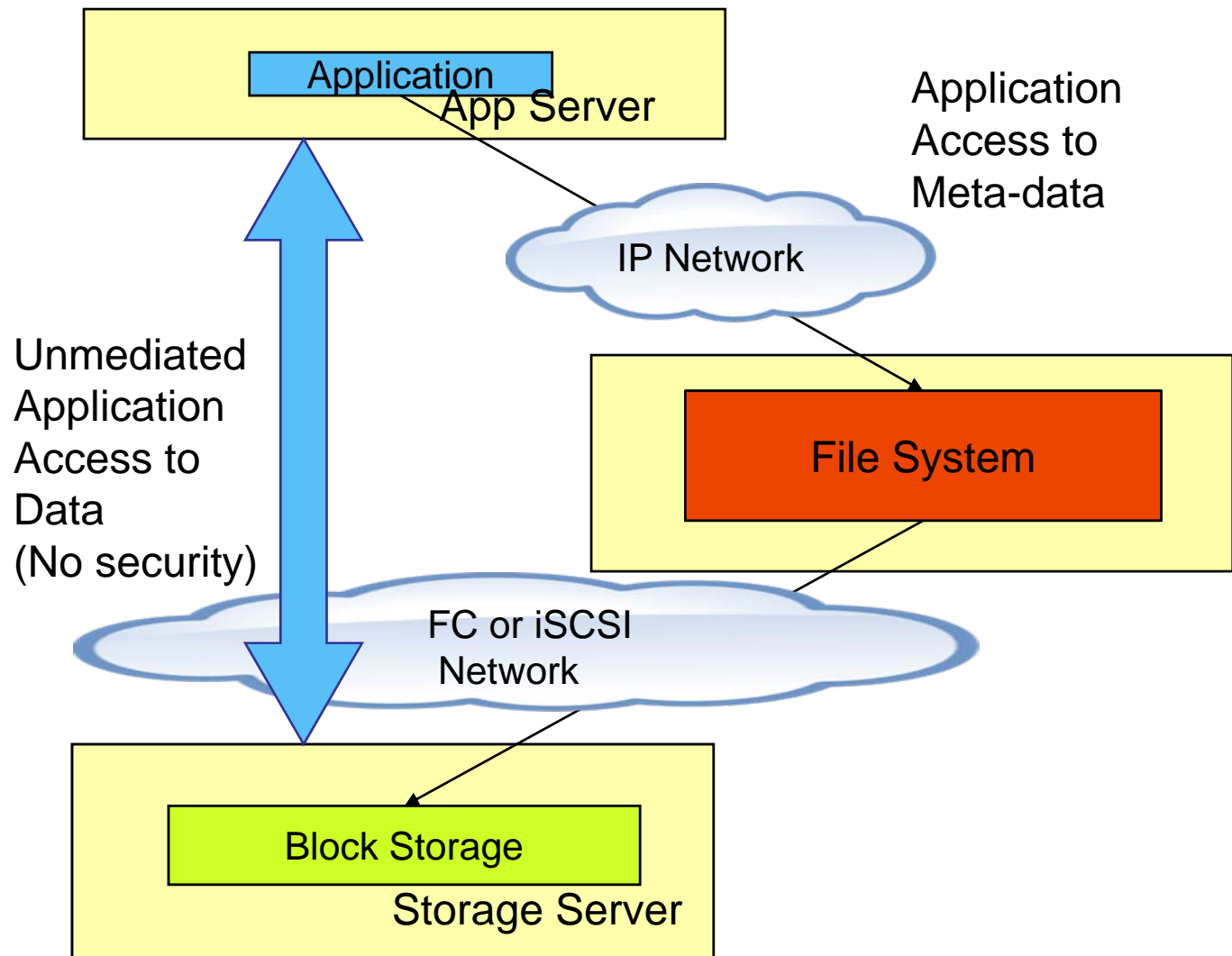
Storage Area Network



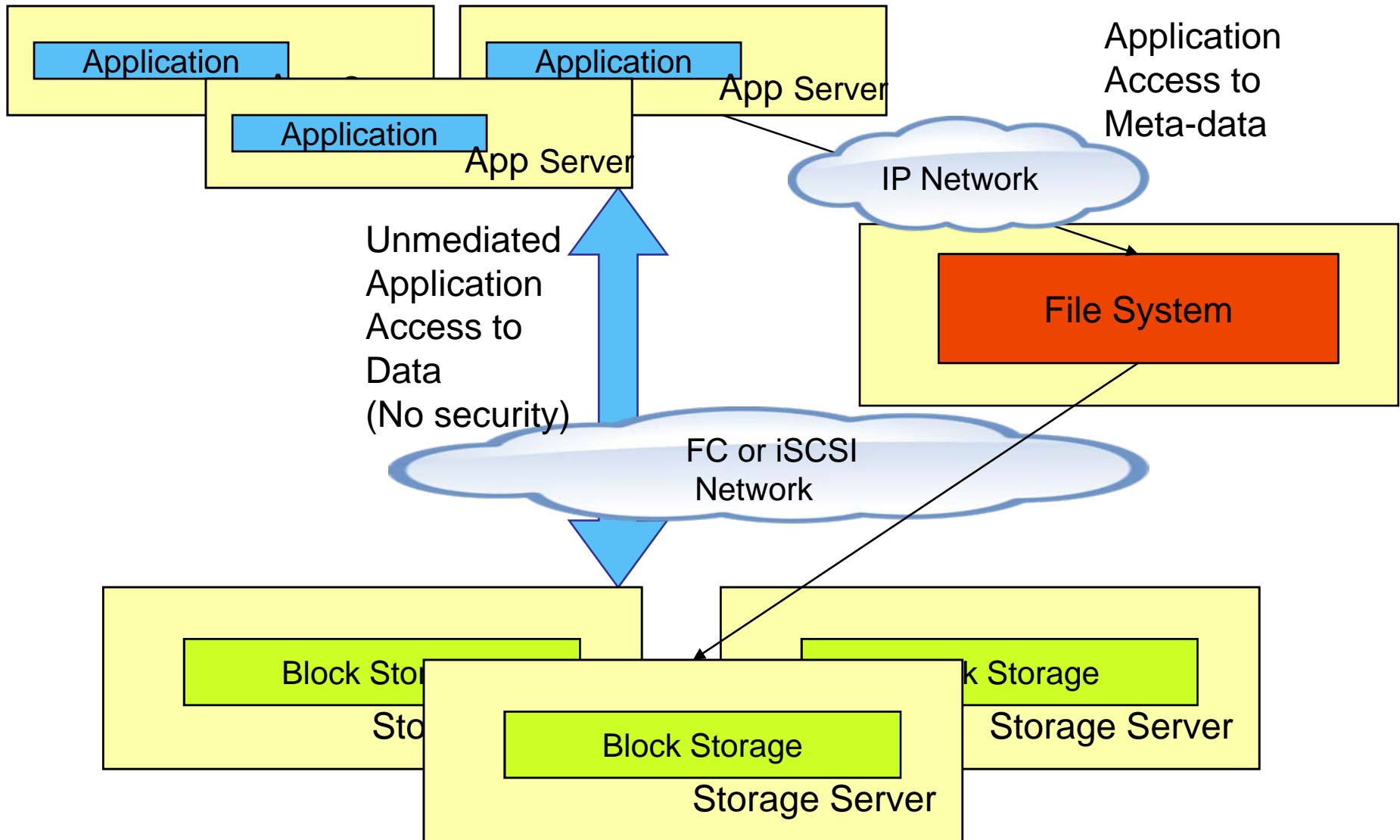
Combined Technologies



SAN Promise – Direct Storage Access



SAN Promise – Disk Sharing



OSDs to the Rescue!

- OSDs can deliver on the promise of SANs while solving the security problems

What is an OSD?

What is an OSD?

- OSD is an acronym for Object Storage Device
- OSDs hold objects, which are like files in a simple file system
 - ◆ Objects are identified by a 64 bit Object ID (OID)
 - ◆ Objects are dynamically created and freed
 - ◆ Object are variable length
 - ◆ Objects in an OSD are grouped within partitions, which are identified by a 64 bit Partition ID
 - 64 bit OID plus 64 bit PID gives a 128 bit namespace
- OSDs manage space allocation of Objects

Capabilities

- Unlike disks, where access is granted on an all or nothing basis, OSDs grant or deny access to individual objects based on *Capabilities*
- A Capability must accompany each request to read or write an object
 - ◆ Capabilities are cryptographically signed by the Security Manager and verified by the OSD
 - ◆ A Capability to access an object is created by the Security Manager, and given to the client (application server) accessing the object
 - ◆ Capabilities can be revoked

OSD -- Like a Disk, but Different

- **Disk storage**
 - ◆ Fixed array of blocks
- **Disk operations**
 - ◆ Read block, write block, format
- **Disk security**
 - ◆ Zoning and LUN Masking of entire disk
- **Transport**
 - ◆ FC SCSI and iSCSI
- **OSD storage**
 - ◆ Many objects of variable size
- **OSD operations**
 - ◆ Read object, write object, create object, list objects, etc.
- **OSD security**
 - ◆ On a per object basis using Capabilities
- **Transport**
 - ◆ FC SCSI, iSCSI, RPC

OSD Advantages over Disks

- Grouping data in objects allows the OSD to know that data in one Object is related and different from data in another Object
 - ◆ Allows the OSD to optimize access to related blocks
- The OSD standard has commands beyond simply storing and retrieving objects:
 - ◆ Snapshots – snapshot a group of Objects (similar to the snapshots in higher end arrays)
 - ◆ Arbitrary attributes – the standard includes attributes on Objects, some of which can be vendor defined to indicate things like Quality of Service, I/O hints, Differentiation of data, etc.

OSD Standards

- There is a standard for OSDs under ANSI INCITS T10 (the SCSI specification)
 - ◆ OSD-1 is basic functionality
 - › Read, write, create, delete objects and partitions
 - › Security model, Capabilities, manage shared secrets and working keys
 - ◆ OSD-2 adds:
 - › Snapshots
 - › Collections of Objects
 - › Extended exception handling and recovery
 - ◆ OSD-3 is in progress:
 - › Device to device communication
 - › RAID-[1,5,6] implemented between devices

OSD Form Factors



Disk array/server subsystem
Example: custom-built HPC systems
predominantly deployed in national labs



Storage bricks for objects
Example: commercial supercomputing offerings

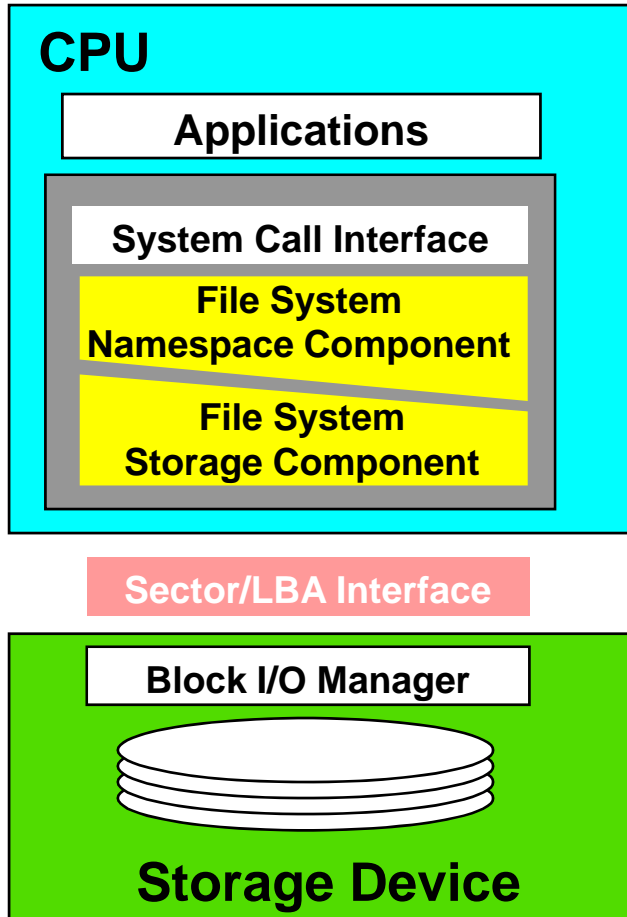


Object Layer Integrated in Disk Drive
Example: only in prototypes

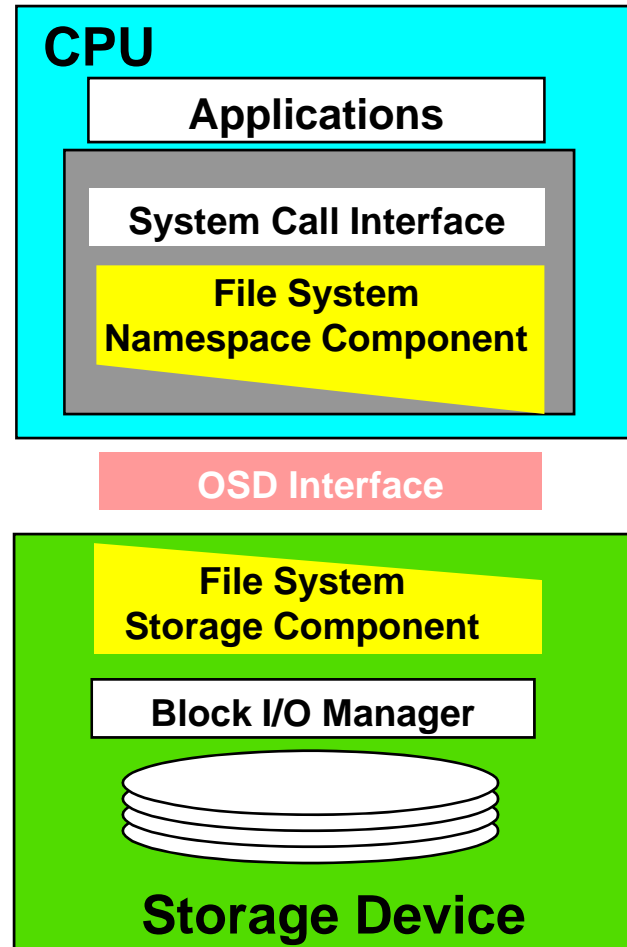
Building File Systems from OSDs

A Simple ObjFS

Traditional Filesystem



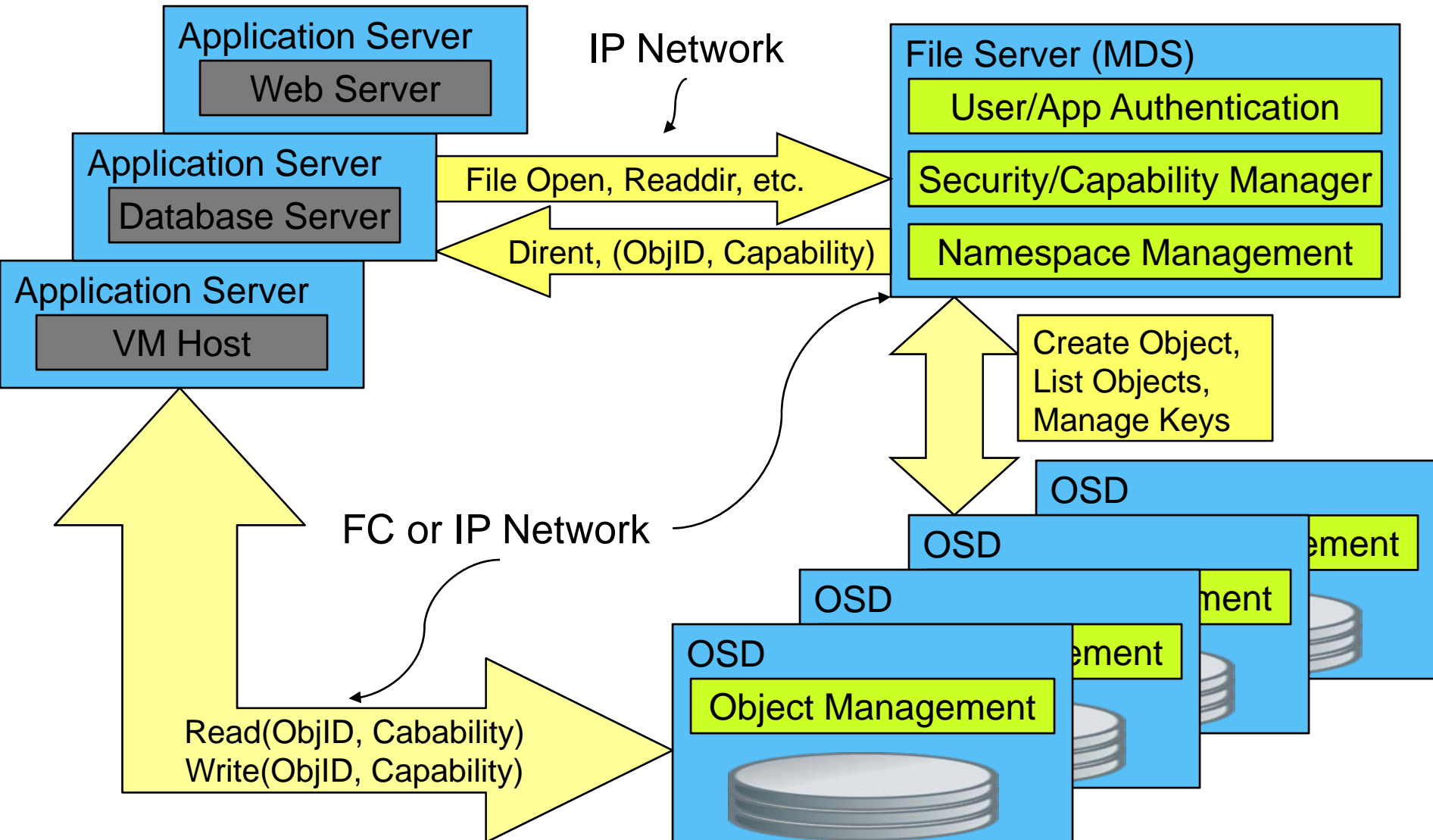
Object-based File System



A simple ObjFS, cont.

- Data for each file is stored in a single object
- File system layer in host manages:
 - ◆ Human readable namespace
 - ◆ User authentication, permission checking, ACLs
 - ◆ OS interface
- Object Layer in OSD manages:
 - ◆ Block allocation and placement
 - ◆ OSD has better knowledge of disk geometry and characteristics so it can do a better job of file placement/optimization than a host-based file system

A More Interesting ObjFS



A More Interesting ObjFS, cont.

- App servers (clients) have direct access to storage to read and write file data securely
 - ◆ Contrast with SAN where security is lacking
 - ◆ Contrast with NAS where server is a bottleneck
- File system includes multiple OSDs
 - ◆ Grow the file system by adding an OSD
 - ◆ Increase bandwidth at the same time!
 - ◆ Can include OSDs with different performance characteristics (SSD, SATA, SAS)
- Multiple File Systems share the same OSDs
 - ◆ Real storage pooling!

A More Interesting ObjFS, cont.

- Allocation of blocks to Objects handled within OSDs
 - ◆ Partitioning improves scalability
 - ◆ Compartmentalized managements improves reliability through isolated failure domains
- The File Server piece is called the MDS
 - ◆ Meta-Data Server
 - ◆ Can be clustered for scalability

RAIOSD– Redundant Array of Inexpensive OSD

- In some implementations, an OSD is implemented on top of one or a few disks
- Disks fail, but RAID techniques can help
 - ◆ Apply RAID at the level of individual Objects!
- OSD RAID levels
 - ◆ RAID-1 (mirroring) – the same data is written to two or more Objects on different OSDs
 - ◆ RAID-5 (parity) – data is written across 4 Objects and parity to a 5th Object on different OSDs
 - ◆ Protects data from the loss of an OSD!

RAIOSD, cont.

- Implementing RAID across Objects allows parity declustering
 - ◆ If an OSD fails, the Objects can be rebuilt across all remaining OSDs
- RAID-0 and RAID-5 also mean files are striped across multiple Objects for higher performance

Production Quality Object-Based File Systems

ObjFS in Production Use

- There are several Object-Based File Systems used in production
 - ◆ In the national labs...
 - › Object-based, but not OSD compliant
Limited motivation to move toward OSD compliance.
 - ◆ Commercial supercomputing-oriented
 - › Object-based, moving toward OSD-2 compliance
 - ◆ pNFS (NFS v4.1)
 - › Includes OSD standard compliant devices for storage
- All have the architecture described in
“A More Interesting ObjFS” (slides 22-24)

Lustre

- Supercomputing focus emphasizing
 - ◆ High i/o throughput
 - ◆ Scalability in the Pbytes of data and billions of files
- OSDs called OSTs (Object Storage Targets)
- Only RAID-0 supported across Objects
 - ◆ Redundancy inside OSTs
- Runs over many transports:
 - ◆ IP over ethernet
 - ◆ Infiniband
 - ◆ Myrinet
 - ◆ Quadric Elan

Lustre, cont.

- OST and MDS are Linux based
 - ◆ Other platforms under consideration
- Client software supports Linux
 - ◆ Other platforms under consideration
- In use in:
 - ◆ Oil & gas
 - ◆ Telecom
 - ◆ Video post-production
 - ◆ Pittsburgh Supercomputing Center
 - ◆ Aerospace
 - ◆ National labs

Blade-oriented ObjFS offerings

- Supercomputing focus emphasizing
 - ◆ High i/o throughput
 - ◆ Scalability to Pbytes of data and billions of files
- OSDs are called Storage Blades
- MDSs are called Director Blades

- RAID-5 supported across Objects
 - ◆ Additional redundancy inside OSDs and over network

Blade-oriented ObjFS offerings

➤ Common transports

- ◆ Infiniband
- ◆ iSCSI over 1 Gbit and 10 Gbit Ethernet

➤ Custom clients: typically Linux

➤ In use in:

- ◆ Mapping
- ◆ National labs
- ◆ Energy
- ◆ UC Berkeley Center for Integrative Genomics
- ◆ UC San Diego Center for Marine Genomics

pNFS – NFS v4.1

- pNFS is a nickname for NFS Version 4.1
- A protocol, not a product
- Supports direct access to storage from a pNFS client
- Three flavors of pNFS with three types of storage:
 - ◆ Objects in an OSD
 - ◆ Files in an NFS file server
 - ◆ Blocks in a disk on a SAN

- pNFS uses IP as a transport, typically over Ethernet
 - ◆ IP over InfiniBand or other transports possible
- Major vendors are supporting pNFS development
- pNFS clients available on Linux and OpenSolaris

- Please send any questions or comments on this presentation to SNIA: trackfilesystems@snia.org

**Many thanks to the following individuals
for their contributions to this tutorial.**

- SNIA Education Committee

**Craig Harmer
Julian Satran,
Rich Ramos
Erik Riedel
Mike Mesnier
Ralph Weber**

Appendix

Further Reference

➤ Academic research

- ◆ www.pdl.cmu.edu
- ◆ www.dtc.umn.edu

➤ Standards work

- ◆ www.snia.org/apps/org/workgroup/osd
- ◆ www.tl0.org/drafts.htm
- ◆ www.ietf.org/dyn/wg/charter/nfsv4-charter.html

➤ Industry research & development

- ◆ www.sun.com/lustre
- ◆ www.opensolaris.org/os/project/nfsv4/
- ◆ www.panasas.com

Motivation

- Storage Access has evolved over time
 - ◆ From DAS (Direct Attached Storage)
 - ◆ to NAS (Network Attached Storage)
 - ◆ to SANs (Storage Area Networks)
- But SANs have not fulfilled their promise